Reconocimiento de formas

Curso 2017-2018

Enzo Ferey - enzo.ferey@alumnos.upm.es

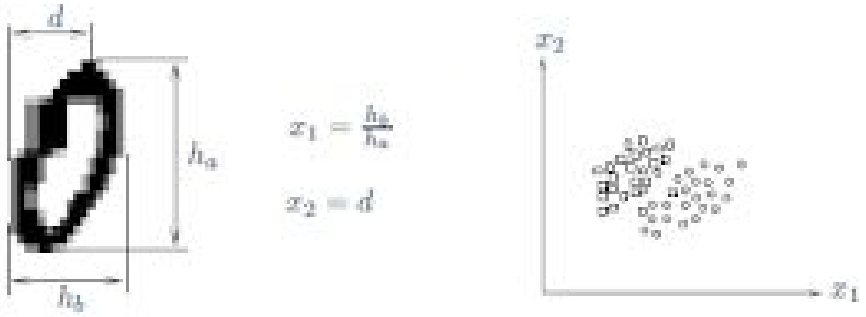
Tema 1 - Introducción

Reconocimiento de formas es un área de la informática en la que se diseñan algoritmos que buscan patrones o regularidades en datos.

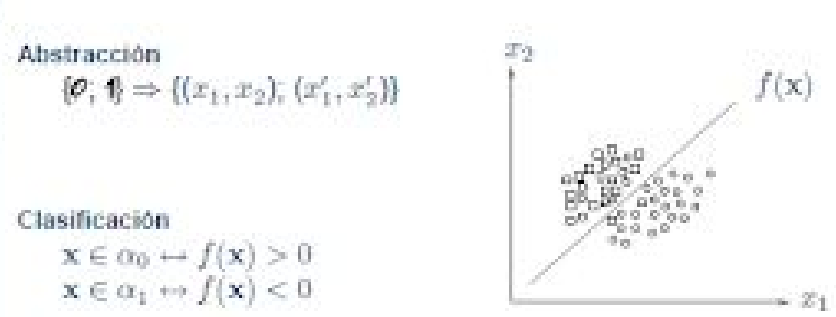
Está relacionada con las palabras más sexys del siglo XXI: machine learning, data mining, big data, artificial intelligence.

Distinguir un 0 y un 1 manuscrito

Se podría por ejemplo plantear medir el trazo del número de la siguiente manera para separar los número en dos grupos:



A continuación se definiría una función f(x) que divide ambos grupos y según si el resultado de la función es mayor o menor que cerro se podrá averiguar de qué número se trata.



Terminología

Universo de trabajo: conjunto de objetos disponibles para la construcción del clasificador.

Clase: agrupación de objetos del universo de trabajo.

Objeto: instancia de una clase.

Vector de características: conjunto de descriptores que representan un objeto y permiten clasificarlo. Se tratan de propiedades discriminantes, fiables, independientes y económicas.

Función discriminante: función que combina las componentes del vector de características y nos permite decidir a qué clase pertenece un objeto. Pueden ser basadas en la regionalización (ejemplo de 0 o 1) o basadas en la distancia (euclídea por ejemplo).

Tipos de clasificadores

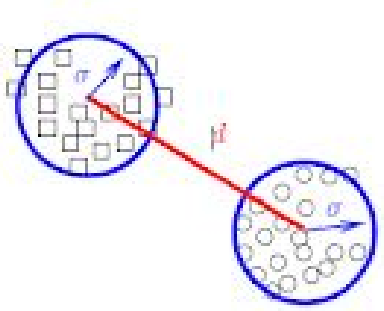
- Supervisados
  Se conoce cuántas clases hay y la pertenencias de los objetos del UT a cada una de ellas. Clasificadores basados en el cálculo de una distancia (tema 2), construyen una función de pertenencia a la clase; o basados en la regionalización (tema 6), construyen directamente la función discriminante que separa las clases.

- No supervisados

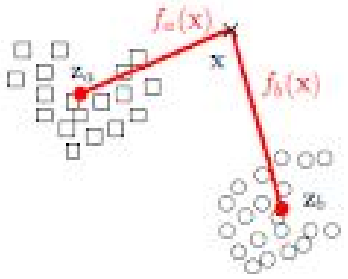
  No se conocen las clases (tema 5)

Clasificador de la distancia euclídea

Se formula la hipótesis de que la dispersión de las clases es pequeña en relación a la distancia entre ellas ($d >> \sigma$).

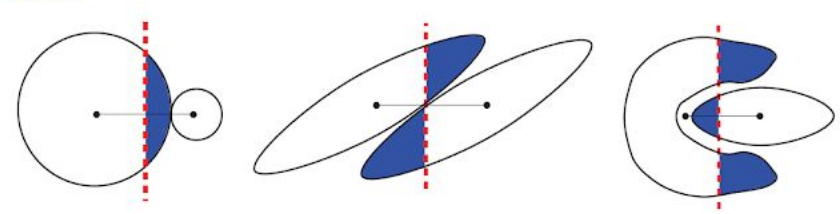


El centroide Z será el representante de cada clase (media de los puntos). A la hora de clasificar un objeto x se calculará su distancia euclídea con todos los centroides de cada clase y se etiquetará como pertenece a la clase del centroide más cercano.



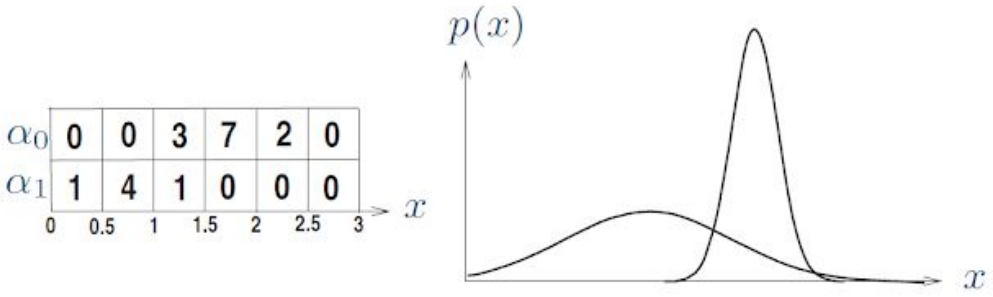
Tema 2 - Clasificador estadístico bayesiano

El clasificador de la distancia euclídea falla cuando no se cumple la hipótesis determinista sobre la distribución de las muestras ya que no modela la distribución de las muestras en cada clase.



En este tema se resuelve dicho problema mediante un método estadístico de la distribución de las muestras de cada clase.

Retomando el ejemplo de los 0 y los 1, vamos a considerar un UT donde hay 12 elementos en las clase del 0 y 6 en la clase del 1.



Distribución de los datos de cada clase

A simple vista de estos datos tendríamos una funciones de pertenencia a cada clase:

$$P(\alpha_0) = \frac{12}{18} = \frac{2}{3}$$

$$P(\alpha_1) = \frac{6}{18} = \frac{1}{3}$$

Y si por ejemplo se nos dice que un objeto x se encuentra entre $1 < x < 1'5$ podríamos decir con seguridad que hay:

$$P(\alpha_0 \mid x) = \frac{3}{4}$$

$$P(\alpha_1 \mid x) = \frac{1}{4}$$

Terminología

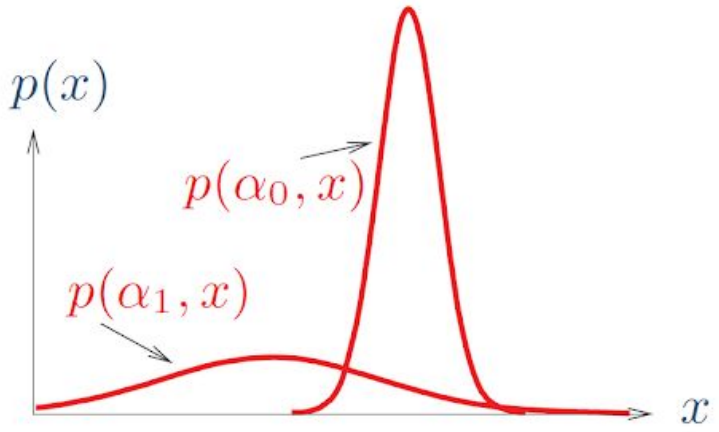
- $p(\alpha_i, x)$: función de densidad de probabilidad conjunta, describe la probabilidad relativa de que dicha variable aleatoria tome el valor x.

$$P(\alpha_i, x) = \frac{h(\alpha_i, a \le x \le b)}{\sum_i card(\alpha_i)}$$

$$p\,(\alpha_i,\ x) = \frac{P(\alpha_i, x)}{\Delta x}$$

Ejemplo con la tabla anterior

$$P(\alpha_0,\ 1'3) = \frac{h(\alpha_0,\ 1 \le x \le 1'5)}{card(\alpha_0) + card(\alpha_1)} = \frac{3}{18} = \frac{1}{6}$$



- $P(\alpha_i)$: probabilidad a priori de la clase α_i

$$P(\alpha_i) = \sum_{\forall x} P(\alpha_i,\ x)$$

Ejemplo con la tabla anterior

$$P(\alpha_0) = \frac{12}{18} = \frac{2}{3}\ ;\ P(\alpha_1) = \frac{6}{18} = \frac{1}{3}$$

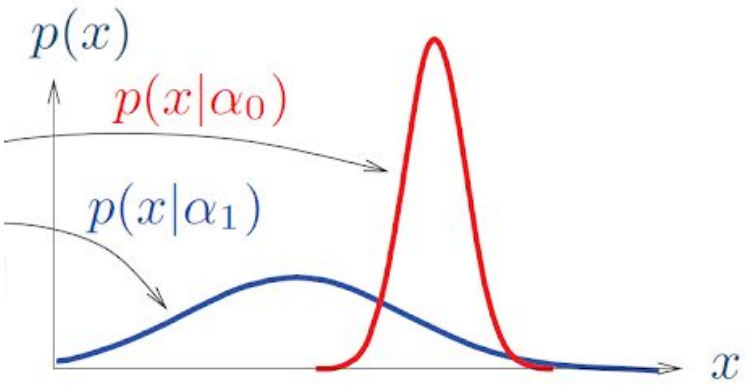
- $p\,(x \mid \alpha_i)$: función de probabilidad condicionada de la clase α_i .

$$P(x \mid \alpha_i) = \frac{h(\alpha_i,\, a \le x \le b)}{card(\alpha_i)}$$

$$p\,(x \mid \alpha_i) = \frac{P(x \mid \alpha_i)}{\Delta x}$$

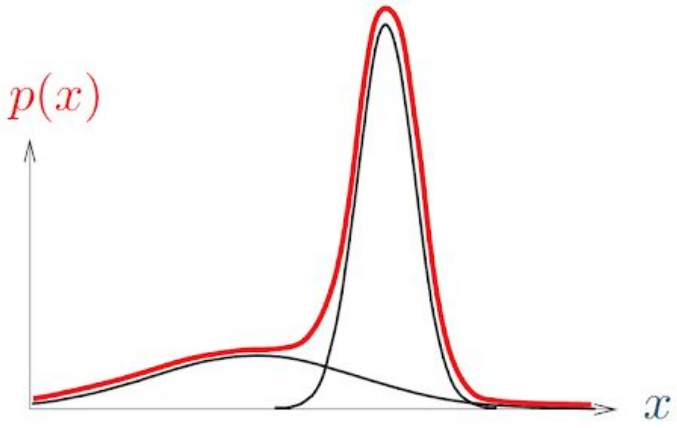
Ejemplo con la tabla anterior

$$P(1'3 \mid \alpha_0) = \frac{h(\alpha_0,\, 1 \le x \le 1'5)}{card(\alpha_0)} = \frac{3}{12} = \frac{1}{4}$$



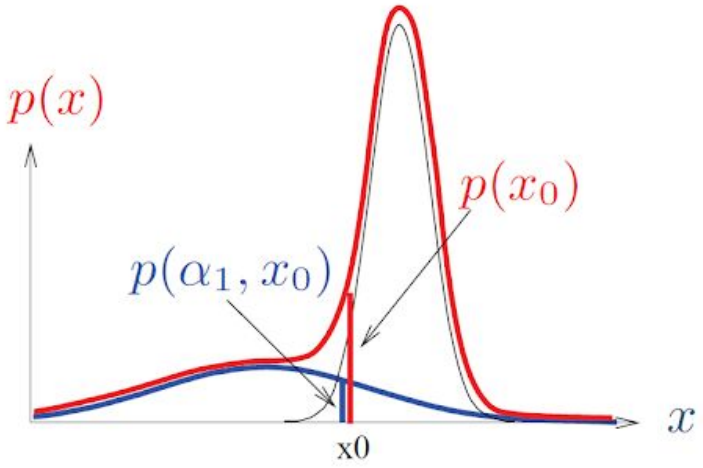
- $p\,(x)$: función de probabilidad marginal de x.

$$p\,(x) \;=\; \sum_{\forall\,\alpha_i} p(x \mid \alpha_i) * P(\alpha_i)$$



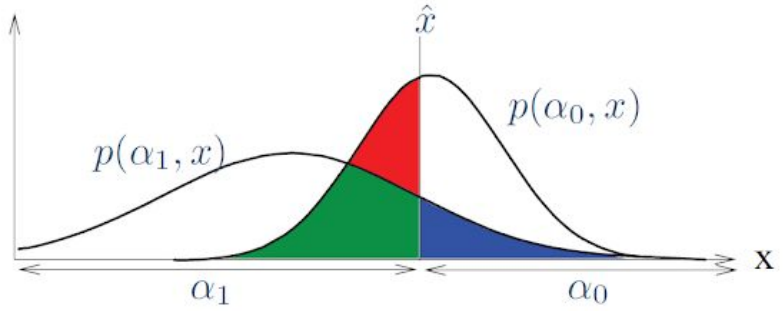
- $P\,(\alpha_i \mid x)$: probabilidad a posteriori de α_i .

$$P(\alpha_i \mid x) = \frac{p(x \mid \alpha_i) * P(\alpha_i)}{p(x)}$$



Clasificación de mínimo error

El objetivo es encontrar funciones de pertenencia asociadas a cada clase que minimicen la probabilidad de error.



$$P(error) = \int_{\mathcal{R}_{\alpha_1}} p(\alpha_0, x)dx + \int_{\mathcal{R}_{\alpha_0}} p(\alpha_1, x)dx$$

El error de la zona verde y la zona azul (significado de la ecuación superior serán constantes, y se busca minimizar el área de la zona roja.

El mínimo está en un x tal que:

- $p(\alpha_1, x) = p(\alpha_0, x)$
- $p(x|\alpha_1)P(\alpha_1) = p(x|\alpha_0)P(\alpha_0)$, ya que $p(\alpha, x) = p(x|\alpha)P(\alpha)$.
- $p(\alpha_1|x) = p(\alpha_0|x)$, ya que según el teorema de Bayes $p(\alpha|x) = \frac{p(\alpha,x)}{p(x)} \propto p(\alpha, x)$

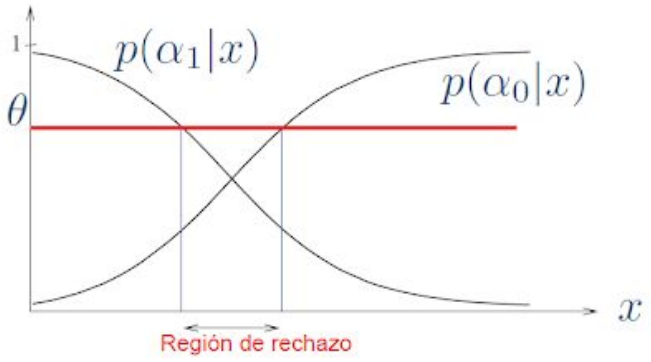
Por lo tanto, el criterio de clasificación que minimiza la probabilidad de error asigna un objeto a aquella clase cuya probabilidad a posteriori sea máxima.

$$f_i(x) = P(\alpha_i \mid x) = p(x \mid \alpha_i) * P(\alpha_i)$$

A la hora de hacer el clasificador da igual usar la probabilidad a posteriori o la conjunta porque una es la otra dividida por un mismo valor.

En algunas aplicaciones es preferible no tomar una decisión antes que cometer un error (por ejemplo en el diagnóstico de un cáncer). Las regiones donde la probabilidad de error es más alta son aquellas en las que $p\ (\alpha_k,\ x)$ tiene valores semejantes $\forall k$.

En esos casos se establecerá un umbral dentro del cual se rechazará cualquier objeto.



Métodos paramétricos de clasificación

Dados unos conjuntos de datos pertenecientes a distintas clases, se quiere construir las funciones de pertenencia de cada clase de la forma $f_i(x) = p(x \mid \alpha_i) * P(\alpha_i)$.

Se supondrá que las muestras del UT son independientes e idénticamente distribuidas y que $p(x \mid \alpha_i)$ tiene una forma paramétrica conocida: $p(x \mid \alpha_i,\ \theta_i)$. Por lo tanto mediante una estimación de máxima verosimilitud se obtendrá un conjunto de ecuaciones para estimar θ.

A partir de ahora $p(x \mid \theta) \equiv p(x \mid \alpha_i,\ \theta_i)$.

Tendremos pues que resolver resolver *n* problemas de estimación paramétrica.

En primer lugar, emitiremos la hipótesis de que los datos siguen una distribución gaussiana para podemos calcular los parámetros de dicha distribución de la siguiente forma.

Supondremos que $p(\mathbf{x}|\boldsymbol{\theta}) \sim \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \Sigma)$. Es decir,

- Si dim(x)=1

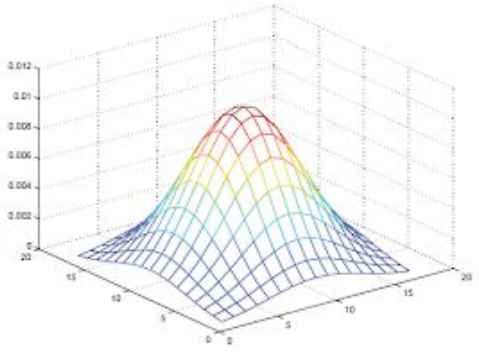
$$p(x|\boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

donde $\mu = \mathrm{E}\{x\}$ y $\sigma = \mathrm{E}\{(x-\mu)^2\}$.

- Si dim(x)=d

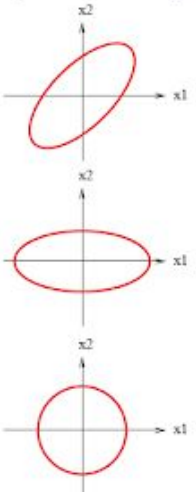
$$p(\mathbf{x}|\boldsymbol{\theta}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

donde $\boldsymbol{\mu} = \mathrm{E}\{\mathbf{x}\}$ y
$\Sigma = \mathrm{E}\{(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^\top\}$.



Dichos parámetros representan la posición y la orientación de la distribución:

Si suponemos que $\mathbf{x} = (x_1, x_2)$, entonces



$$\Sigma = \begin{pmatrix} \sigma_1^2 & \mathrm{cov}(x_1, x_2) \\ \mathrm{cov}(x_2, x_1) & \sigma_2^2 \end{pmatrix}, \sigma_i^2 = \mathrm{var}(x_i)$$

$$\Sigma = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} \sigma & 0 \\ 0 & \sigma \end{pmatrix} \equiv \sigma \mathbf{I}$$

A continuación a partir de las fronteras de indecisión se obtendrán las funciones de pertenencia.

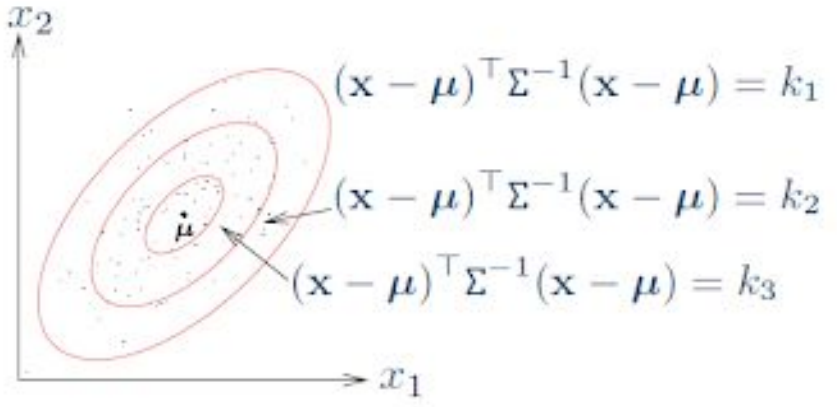
Dadas dos clases α_i y α_j , la frontera de indecisión es el lugar geométrico del espacio tal que $f_i(x) = f_j(x)$. En general, para el caso biclase, la frontera es una hipercuádrica.

Se pueden dar tres casos (Σ = covarianzas):

- Caso general ($\Sigma_i \neq \Sigma_j$): distinta forma. Se utiliza cuando hay muchos datos.
- Clases isomorfas ($\Sigma_i = \Sigma_j$): misma forma. Se utiliza cuando hay pocos datos.
- Clases isomorfas e isotrópicas ($\Sigma_i = \Sigma_j = \sigma^2 I$): misma forma e igual dispersión en todas las direcciones del espacio. Se utiliza cuando hay muy pocos datos. En este caso el clasificador es equivalente al de la distancia euclídea.

La distancia de mahalanobis es la dependencia funcional de la gaussiana sobre x. Puede verse como una generalización de la distancia euclídea en la que se tiene en cuenta la dispersión de las clases.

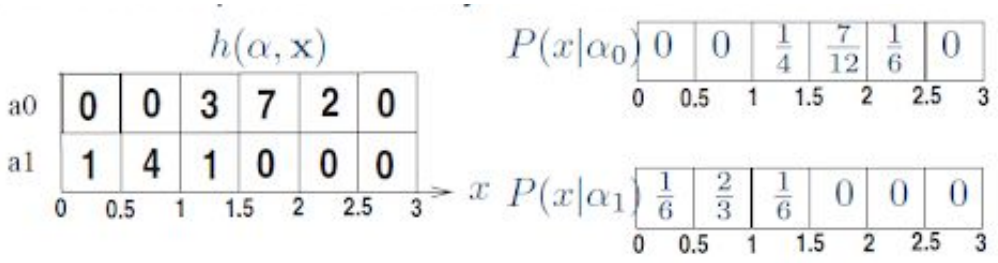
La densidad de probabilidad de la gaussiana es constante en las regiones del espacio tal que $d_M(x) = k$.



Métodos no paramétricos de clasificación

Esto consiste en representar $p(x \mid \alpha)$ sin necesidad de conocer la forma paramétrica de ésta ($p(x) \equiv p(x \mid \alpha)$).

Retomando el ejemplo anterior se divide el espacio de trabajo en volúmenes constantes:



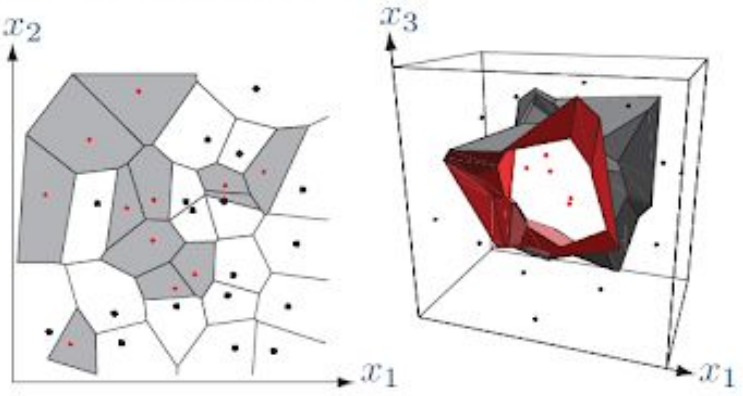
El método de k-vecinos establece un valor constante para $n_{v(x)} = k$. $V(x)$ representa el menor volumen entorno a x que contiene k muestras.

$$\hat{p}(x) = \frac{k}{n * V(x)}$$

$$f_i(x) = \hat{p}(x|\alpha_i) * P(\alpha_i) = \frac{k_i}{n_i * V(x)} * \frac{n_i}{n}$$

Siendo *k* un parámetro constante a fijar (si es pequeño la estimación es irregular y si aumenta se vuelve más homogénea) y *n* siendo también constante, la función de pertenencia vendrá dada por el número de muestras de la clase α_i en sus *k* vecinos.

Para k = 1 se obtiene el clasificador del vecino más próximo, el cual en dos dimensiones da lugar a una segmentación del espacio en células de Voronoi.

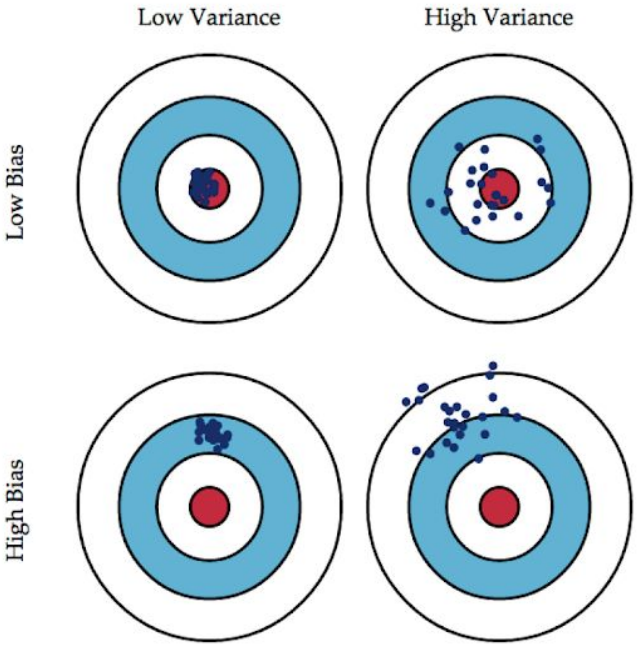


Cuando *n* es grande su error es como máximo el doble del óptimo.

Tema 3 - Evaluación del rendimiento

Una vez construido un clasificador es necesario evaluar su rendimiento y compararlo con otros clasificadores. Preferimos aquel que mejor generalice, esto es el que mejor clasifique muestras que no se emplearon para entrenarlo.

Las dos propiedades claves del rendimiento de un clasificador son su **dispersión** (bias) y su **varianza** (variance).



El método habitual consiste en cambiar el número de parámetros y comparar dichas propiedades hasta llegar a un balance de ambas.

Algunas medidas para reducir la varianza son: agrandar el dataset, reducir el número de parámetros e incrementar la regularización.

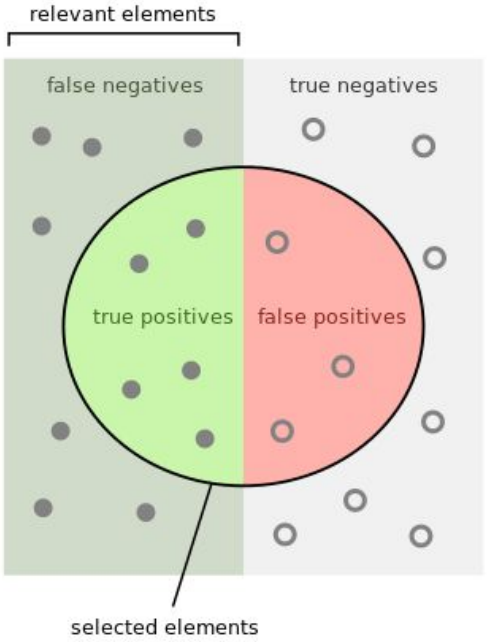
Algunas medidas para reducir la dispersión son: incrementar el número de parámetros y reducir la regularización.

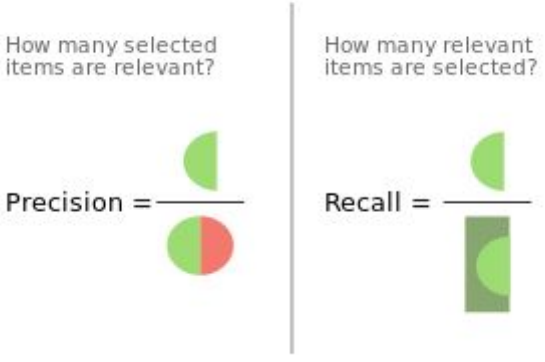
También habrá que tener en cuenta con entrenar demasiado (**overfitting**) o no entrenar suficiente (**underfitting**) nuestro clasificador.

Un alto número de errores en el entrenamiento y en las pruebas indica underfitting y un alto número de errores en las pruebas y bajo número de errores en el entrenamiento indica overfitting.

Medidas de estimación del rendimiento

- Tasa de error: consiste en calcular el porcentaje de acierto en la clasificación. Es una medida totalizador que no discrimina cómo se distribuyen los errores.
- Matriz de confusión: matriz que indica el número de muestras de una clase clasificadas como otra (o la misma). Cada fila es la clase predicha y cada columna la verdadera clase.
- Curva ROC: permite diferenciar verdaderos positivos frente a falsos positivos.
- Precision-recall: permite saber, a partir de los resultados de la Curva ROC nuestro porcentaje de elementos relevantes.





Procedimientos de estimación de tasa de error

- Resustitución: se utiliza el mismo conjunto de datos para entrenar y para evaluar el clasificador.
- Exclusión: se entrena con un subconjunto de los datos A y se evalúa con un subconjunto B. No hay elementos de A que estén en B y viceversa. La suma de los dos subconjuntos representa todo el conjunto de datos.
- Validación cruzada: se divide el conjunto de datos en k grupos del mismo tamaño cuyas intersecciones sean nulo. Se reserva un grupo para la evaluación y se usa el resto para el entrenamiento. La tasa de error será la media de los errores de cada clasificador entrenado dejando cada uno de los grupos para la evaluación.

Bootstrap

Técnica de remuestreo que permite calcular una distribución $p(\theta)$ sobre la estimación de un parámetro θ a partir de un conjuntos de muestras D. Se utiliza para estimar por el ejemplo el sesgo de una muestra.

Test de McNemar

Determina si la diferencia en el rendimiento de dos clasificadores es estadísticamente significativa.

Sean C_a y C_b dos clasificadores, y sean

- n_{00} # muestras clasificadas erróneamente por C_a y C_b
- n_{01} # muestras clasificadas erróneamente por C_a y no por C_b
- n_{10} # muestras clasificadas erróneamente por C_b y no por C_a.

El estadístico z^2, siendo

$$z = \frac{|n_{01} - n_{10}| - 1}{\sqrt{n_{10} + n_{01}}},$$

se distribuye aproximadamente como una χ^2 con un grado de libertad. La hipótesis nula (ambos clasificadores tienen el mismo rendimiento) se puede rechazar (con probabilidad de error 0.05) si $|z| > 1{,}96$.

Tema 4 - Reducción de la dimensionalidad

Nuestro objetivo en este tema es aprender a preprocesar el conjunto de datos de entrada eliminando datos atípicos y aquellos que proporcionen información redundante así como normalizar los vectores de características.

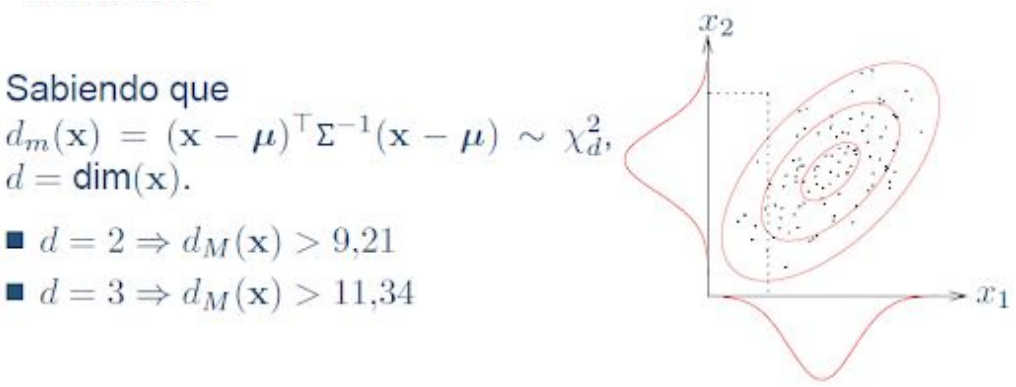
Esto será especialmente útil para resolver problemas de dimensionalidad, disminuir el coste computacional de los cálculos posteriores y poder visualizar los datos más fácilmente.

Eliminación de datos atípicos

Suponiendo que las clases son unimodales y aproximadamente gaussianas, se elimina todo x_i tal que $\exists j$ tal que:

$$\frac{\left|x_{ij} - median(x_{:j})\right|}{median(\left|x_{:j} - median((x_{:j})\right|)} > 4'5$$

Sin embargo hay valores atípicos que no son detectado en la búsqueda univariante y habrá que recurrir a la la distancia de mahalanobis para encontrarlos.



Los valores de Σ y μ se ven afectados por los datos atípicos, por lo que se suele repetir el proceso de estimación de Σ, μ y el de eliminación.

Eliminación de variables correladas

Cuando hay características correladas se puede decir que existe una dependencia lineal entre ellas y por lo tanto una es redundante.

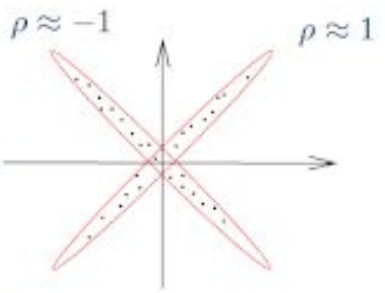
Sea $\mathcal{V}$ la matriz diagonal

$$\mathtt{V} = \begin{pmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2, \end{pmatrix}$$

la matriz de correlaciones vendrá dada por

$$\rho = \mathtt{V}^{-\frac{1}{2}} \Sigma \mathtt{V}^{-\frac{1}{2}}.$$

Si $|\rho(x_i, x_j)| \geq 0{,}95$, elimino una de las características.



Normalización

Estandarizando las variables conseguimos que todas ellas tengan media cero y varianza unidad, de esta forma se dispone de características cuyas magnitudes numéricas sean comparables.

Se puede normalizar cada variable individualmente o el vector de características entero.

- Estandarizando cada variable individualmente

$$\mu_j = \frac{1}{m}\sum_{j=1}^{m} x_{ij}; \quad \sigma_j^2 = \frac{1}{m-1}\sum_{j=1}^{m}(x_{ij} - \mu_j)^2;$$

  de donde $\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$.

- Estandarizando el vector de características
  Sea $\Sigma = \mathtt{CDC}^\top$ la descomposición espectral de Σ, y sea $\Sigma^{-\frac{1}{2}} = \mathtt{CD}^{-\frac{1}{2}}$.
  La transformación de estandarización: $\hat{\mathbf{x}} = \Sigma^{-\frac{1}{2}}(\mathbf{x} - \boldsymbol{\mu})$.

Reducción de dimensionalidad del vector característico

Dado un vector de características con n componentes se trata de encontrar el subconjunto que mejor discrimina tal que su cardinalidad sea $k < n$.

El problema se resuelve optimizando un criterio de selección $J(V)$. Este criterio podrá ser una técnica de Wrapper (mide el rendimiento del clasificador) o bien una técnica de Filter (mide la separabilidad).

Para encontrar el subconjuntos de características en cuestión se realizarán distintos métodos de búsqueda:

- Métodos óptimos
    - **Búsqueda exhaustiva**: se estudia todas las combinaciones de *n* características tomadas en conjuntos de *k* muestras.
    - **Branch and bound**: se recorre el mismo árbol que en método anterior pero esta vez se para de explorar una rama si se obtiene un valor del criterio de selección menor que el del nivel superior.

- Métodos no óptimos
    - **Evaluación individual**: se aplica el criterio a cada características y se seleccionan las k mejores. No considera relaciones multivariantes.
    - **Inclusión secuencial**: parte del conjunto vacío y en cada paso *k* incluye la característica *c* que maximiza el valor del criterio $J(W_k \cup \{c\})$.
    - **Exclusión secuencial**: parte del conjunto total y en cada paso *k* elimina la característica *c* que maximiza el valor del criterio $J(W_k - \{c\})$.
    - **Búsqueda flotante**: permite eliminar y añadir características considerando en cada iteración cualquiera de las pertenencias a W_k y a $V - W_k$.

Para medir solapamiento entre las clases (similaridad) se utilizarán técnicas paramétricas como la divergencia, distancia de Bhattacharyya y distancia de Mahalanobis y técnicas no paramétricas como el área bajo la curva ROC y el ratio de Fisher.

(Paramétrico = las ecuaciones de la curva vienen dadas respecto a un parámetro)

Análisis de Componentes Principales (ACP) - No supervisado

Dado X, encontrar *A* tal que $X_{mxn}\, A_{nxk} = Z_{mxk}$ donde $k < n$ y Z es la mejor aproximación de X de dimensión k. Esto es equivalente a buscar un *A* que minimice la suma de los errores de reconstrucción y maximice la varianza de la variable transformada.

Maximiza la varianza de las nuevas características discriminantes.

El ACP busca la proyección según la cual los datos queden mejor representados en términos de mínimos cuadrados. Esta convierte un conjunto de observaciones de variables posiblemente correlacionadas en un conjunto de valores de variables sin correlación lineal llamadas componentes principales.

Para ello se construye una transformación lineal que escoge un nuevo sistema de coordenadas para el conjunto original de datos en el cual la varianza de mayor tamaño del conjunto de datos es capturada en el primer eje (llamado el Primer Componente Principal), la segunda varianza más grande es el segundo eje, y así sucesivamente.

Para construir esta transformación lineal debe construirse primero la matriz de covarianza o matriz de coeficientes de correlación. Debido a la simetría de esta matriz existe una base completa de vectores propios de la misma. La transformación que lleva de las antiguas coordenadas a las coordenadas de la nueva base es precisamente la transformación lineal necesaria para reducir la dimensionalidad de datos.

Algoritmo

Entrada: $\mathcal{X}$, m vectores de características por columnas.
Salida: $\mathtt{A}$, matriz de proyección buscada. $\bar{\mathtt{x}}$, media de los datos de entrada.

1. Calcular $\bar{\mathtt{x}} = \frac{1}{m}\sum_{i=1}^{m} \mathtt{x}_i$
2. Calcular $\hat{\mathcal{X}} = \mathcal{X} - \bar{\mathtt{x}}$.
3. Calcular $\Sigma_{\mathcal{X}}$, la matriz de covarianzas de $\hat{\mathcal{X}}$.
4. Calcular la descomposición espectral (en autovalores y autovectores) de $\Sigma_{\mathcal{X}} = \mathtt{P}\Lambda\mathtt{P}^{\top}$ (`eig` en octave/matlab).
5. La matriz de proyección será $\mathtt{A} = \mathtt{P}_{1-k}$ (donde $\mathtt{P}_{1-k}$ es la matriz cuyas columnas son los k autovectores con mayor autovalor).

Transformación PCA de $\mathtt{x}$: La transformación de un nuevo vector columna $\mathtt{x}$, vendrá dada por $\mathtt{z} = \mathtt{A}^{\top}(\mathtt{x} - \bar{\mathtt{x}})$.

Análisis Discriminante Lineal (ADL) - Supervisado

Su objetivo es encontrar una combinación lineal de rasgos que caracterizan o separan dos o más clases.

Dado X, encontrar *A* tal que $X_{mxn}\, A_{nxk} = Z_{mxk}$ donde $k < n$ y en Z se maximiza el Ratio de Fisher. Se utiliza distancia entre las medias normalizada por la dispersión de cada clase.

Maximiza el índice de separabilidad de clases en el espacio transformado.

Algoritmo

Entrada: $\mathcal{X}$, m vectores de características por columnas. $\mathbf{y}$ vector de enteros con las C etiquetas de clase.
Salida: $\mathtt{A}$, matriz de proyección buscada.

1. Calcular $\mathtt{S}_d$, matriz de dispersión dentro de la clase con $\mathbf{y}$ y $\mathcal{X}$.
2. Calcular $\mathtt{S}_b$, matriz de dispersión entre clases con $\mathbf{y}$ y $\mathcal{X}$.
3. Calcular la descomposición espectral de $\mathtt{S}_d^{-1}\mathtt{S}_e = \mathtt{P}\Lambda\mathtt{P}^\top$ (*eig* en octave/matlab).
4. La matriz de proyección será $\mathtt{A} = \mathtt{P}_{1-(C-1)}$ (donde $\mathtt{P}_{1-(C-1)}$ es la matriz cuyas columnas son los $C-1$ autovectores con mayor autovalor).

Transformación LDA de $\mathbf{x}$: La transformación de un nuevo vector columna $\mathbf{x}$, vendrá dada por $\mathbf{z} = \mathtt{A}^\top\mathbf{x}$.

Tema 5 - Clustering

Se posee un conjunto de datos del que no se conoce las etiquetas ni patrones y se agrupan de forma no supervisada.

Algoritmo K-means

Dada una estimación inicial del número k de grupos:

1. Inicializar centroides de cada clase de forma aleatoria
2. Bucle hasta convergencia:
   2.1. Calcular la distancia euclídea de todos los datos respecto al centroide de cada grupo.
   2.2. Clasificar cada dato en el grupo cuya distancia sea menor.
   2.3. Actualizar centroides con la nueva clasificación de los datos.

También se puede optimizar la inicialización separando los clusters iniciales lo máximo posible (distancia euclídea o distancia de Mahalanobis).

Algoritmo Sequential

Dado un umbral por defecto y un número máximo de clusters:

1. Se toma el primer elemento del dataset como centroide del primer cluster.
2. Repetir para elemento restantes (índices 2...n):

a. Calcular la distancia euclídea de todos los puntos con todos los centroides existentes y tomar menos distancia de todas las combinaciones.
    i. Si esa distancia es mayor que el umbral se crea un nuevo cluster con ese elemento como centroide.
    ii. En caso contrario añadir dicho elemento al cluster más cercano.
b. Actualizar centroides de clusters que hayan cambiado.

Índice compactación

El índice de compactación de un cluster es el porcentaje de elementos que son de la clase de moda en el cluster.

El índice global de compactación es la media de los índices de compactación de cada cluster.

Un índice de 100% de compactación indica que en un cluster en particular todos los elementos tienen la misma etiqueta (lo cual significa que se ha etiquetado perfectamente). Nota, esto no significa necesariamente que los clusters están separados lo suficiente entre ellos. Es por ello que el índice real de un buen clustering vendrá dado por el ratio de Fisher o el índice de Dunn.

Ratio de Fisher

El ratio de fisher es indicativo de la calidad de todo el conjunto de clusters.

Se calcula como la media de la distancia euclídea entre los centroides de cada uno de los clusters dividido entre la media de la desviación estándar de los datos de un cluster (separación entre cluster / separación del cluster).

Cuando más alto sea su valor más eficiente es el método de clustering y mejores son las variables discriminantes.

Índice de Dunn

Es una alternativa al ratio de Fisher, aunque es más costoso en tiempo de ejecución.

Se calcula dividiendo la mínima distancia entre los puntos de todos los clusters entre la máxima distancia entre dos puntos de un cluster (min distance intercluster / max intracluster dispersion).

Pre-procesamiento de datos

En reconocimiento supervisado puede ser de interés en algunos casos pre-procesar los datos, clase a clase, para mejorar la eficiencia de los clasificadores:

- Normalización min-max: v´= (v- v min) / (v max – v min)
- normalización estadística v´= (v- media) / desv standard

En reconocimiento no supervisado es muy arriesgado pre-procesar y escalar los datos, siendo más habitual y recomendable experimentar con diferentes distancias (Manhattan, coseno…) o con diversas medidas de similitud, tales como las normas de Minkowski L_k .

Tema 6 - Aprendizaje

En este tema se aborda el diseño de clasificadores y reconocedores mediante algoritmos de aprendizaje.

Algoritmo del perceptrón

El objetivo es encontrar una función discriminante para cada una de las clases presentes.

Previamente al entrenamiento se habrá hecho una separación k-fold para hacer pruebas del rendimiento del aprendizaje. Igualmente, el dataset de entrada a la fase de entrenamiento, se dividirá en dos (80-20 / 90-10) para tener un grupo de validación interna para la condición de parada.

Algoritmo de entrenamiento:

1. Se evalúa el dato k del dataset con todas las funciones discriminantes
2. Dado que se conoce la etiqueta de dicho dato, se comprueba si la función discriminante de su clase es el que devuelve mayor valor.
    2.1. Si se encuentra alguna función discriminante que devuelva mayor valor que la de su clase se reduce por un coeficiente multiplicado por el elemento y se aumenta la función discriminante de la clase del datos por la misma cantidad. Nota, aunque se puedan decrementar muchas funciones únicamente se incrementa la de la clase una vez por iteración k.
    2.2. En caso contrario se salta a la siguiente iteración k.
3. La condición de parada será la convergencia hacia un valor de acierto en la clasificación del subconjunto de validación interna relativamente bajo.

La detección automática de la condición de parada es de elevada dificultad, por lo que se suele monitorizar visualmente la convergencia para terminar manualmente la fase de entrenamiento.

Ensemble de reconocedores

Como dato empírico se puede observar que en general M clasificadores independientes tienen mejor rendimiento que un único clasificador. La cuestión es cómo unir esas M opiniones individuales en un pensamiento colectivo. Lo primero que viene a la cabeza es una regla de voto por mayoría, aunque también se dice que muchos necios juntos llegarán a mejores conclusiones que un sabio solo.

Las únicas condiciones es que cada uno de los clasificadores del conjunto tengan por separado una tasa de acierto mayor que el 50% (lo cual sería equivalente a tirar una moneda) y sea independiente del resto.

Algoritmo:

1. Para cada dato aplicarle cada clasificador del ensemble
2. Etiquetarlo con la etiqueta más frecuente.

En cada ensemble además se podrá construir el conjunto de entrenamiento de dos formas distintas:

- Bagging: el conjunto de entrenamiento de cada miembro del ensemble se obtiene con un muestreo aleatorio independiente y con reemplazo del dataset completo.

- Boosting: el conjunto de entrenamiento de cada miembro del ensemble se obtiene mediante un proceso secuencial. Este proceso consiste en:

    1) Entrenar el primer clasificador C con todo el dataset → C1
    2) Hacer pruebas del clasificador C1 con todo el dataset y obtener un grupo de datos mal clasificados D1
    3) Entrenar el segundo clasificador C con el grupo D1 → C2
    4) Hacer pruebas del clasificador C2 con todo el dataset y obtener un grupo de datos mal clasificados D2
    5) Entrenar el tercer clasificador C con el grupo D1 ∩ D2 → C3
    6) ……..

Reconocimiento basado en reglas granuladas

Un clasificador basado en reglas granuladas utilizará reglas de la forma:

[Si cierta condición de las variables discriminantes se cumple] → Etiquetar como clase X

Dichas variables discriminantes tendrán que ser granuladas en tres intervalos de valores: “L” (low), “M” (middle), “H” (high). Los valores que serán tomados como umbrales son:

- $X_i \leq m_i - k\sigma_i \rightarrow$ low
- $m_i - k\sigma_i \leq X_i \leq m_i + k\sigma_i \rightarrow$ middle
- $X_i \geq m_i + k\sigma_i \rightarrow$ high

(m = media, k = parámetro de entrada al proceso, sigma = desviación estándar)

A continuación se generará todas las permutaciones de los 3 valores posible para cada columna del vector de características para obtener todas las reglas. Ese conjunto se tendrá que duplicar para cada clase. Por tanto en un vector de 4 características y 3 clases tendremos 3^4 permutaciones * 3:

(L, L, L, L) → clase 1 (L, L, L, M) → clase 1 …….. (H, H, H, M) → clase 1 (H, H, H, H) → clase 1	(L, L, L, L) → clase 2 (L, L, L, M) → clase 2 …….. (H, H, H, M) → clase 2 (H, H, H, H) → clase 2	(L, L, L, L) → clase 3 (L, L, L, M) → clase 3 …….. (H, H, H, M) → clase 3 (H, H, H, H) → clase 3

Tras esto se puede pasar a evaluar cada una de las reglas con el dataset. Esto es clasificar los datos que cumplan sus condiciones y luego medir qué porcentaje de ese etiquetado ha sido correcto.

Con estos datos se construirá un ranking de las mejores reglas que pasarán a forma el grupo final de reglas del clasificador que se evaluará su rendimiento por k-fold (si tenemos un grupo grande datos) o por “leaving-one-out” (si tenemos un dataset pequeño).

Aunque por lo general las variables discriminantes con las que se trabaja son continuas, en algunos casos podrán ser discretas. En este caso si las variables vienen representadas por intervalos se puede seguir aplicando el método explicado anteriormente, pero en el caso de que vengan expresados por categoría habrá que aplicar un algoritmo de clustering para obtener los grupos granulados en lo que se dividirán los valores.

Aprendizaje semi-supervisado

Hasta ahora el diseño y validación de cualquier reconocedor supervisado pasaba por recoger muestras de datos clasificados en los que el número de miembros a cada clase fuera similar para a continuación entrenar el clasificador según cierto algoritmo (distancia euclídea, estadística o aprendizaje) y posteriormente medir su rendimiento con técnicas de cross validation.

Sin embargo obtener conjuntos de datos etiquetados es una tarea difícil ya supone que alguien los ha etiquetado a mano previamente.

Así, los métodos de aprendizaje semi-supervisados plantean mejorar el rendimiento de los reconocedores supervisados utilizando todos esos infinitos datos no clasificados (aprendizaje permanente).

La idea principal tras éstos es partir de un reconocedor supervisado inicial entrenado por un dataset etiquetado, posteriormente clasificar datos no etiquetados y finalmente añadir dichos nuevos datos al conjunto de entrenamiento para intentar mejorar el porcentaje de acierto.

Podremos diferenciar tres tipos de métodos:

- Self-training: el estándar y descrito anteriormente.
- Co-training: igual que el self-training pero el etiquetado de los nuevos datos se realizará con un ensemble de reconocedores.
- Active learning: método en el cual el propio reconocedor selecciona datos nuevos para que el diseñador los etiquete.

Las tres parámetros fundamentales a tener en mente a la hora de construir un reconocedor semi-supervisado son:

- El número de elementos en el conjunto de entrenamiento inicial: cuanto mayor sea el conjunto inicial menor error habrá al principio.
- El número de elementos no etiquetados: cuando mayor sea mayor convergencia habrá en el número de errores.
- La calidad del etiquetado del reconocedor: es fundamental que el etiquetado sea lo óptimo posible para no entrenar el reconocedor con datos erróneos. Es por ello que se suele utilizar ensembles para el etiquetado de nuevos datos.

El mayor dilema que el diseñador de un reconocedor semi-supervisado se preguntará es ¿añadir este nuevo dato va a mejorar el rendimiento del reconocedor?

Por intuición se tiende a pensar que cuantos más datos (clasificados correctamente evidentemente) mejor será el rendimiento. Sin embargo, ¿será mejor partir de un conjunto de datos inicial gigantesco o más pequeño?

Por norma general se va a obtener un mejor resultado construyendo un reconocedor supervisado inicial lo más robusto posible. Y sinceramente un reconocedor semi-supervisado únicamente merece la pena cuando no se posee suficientes datos para entrenar uno supervisado robusto.